--cerinta proiect 12

--operatie join pe cel putin 4 tabele

--Să se afişeze informaŃii despre turiştii care au achiziŃionat excusii de 2 persoane de la

--agenŃii din ConstanŃa.

select t.nume

from agentie a join excursie e on (a.id\_agentie = e.cod\_agentie )

join achizitioneaza ac on (ac.cod\_excursie = e.id\_excursie)

join turist t on (t.id\_turist = ac.cod\_turist)

where e.nr\_locuri = 2 and initcap(a.oras)= 'Constanta'; --filtrare la nivel de linie

filtrare la nivel de linii

--12. Să se obŃină lista turiştilor care au achiziŃionat excursii cu o durata mai mare de 10

--zile. (se va lua în considerare data\_start şi data\_end)

select t.nume, t.prenume

from turist t

where t.id\_turist in (select cod\_turist

from achizitioneaza ac

where cod\_excursie

in (

select id\_excursie

from excursie

where durata >=10

)

);

select t.nume, t.prenume

from turist t

where t.id\_turist in (select cod\_turist

from achizitioneaza ac

where data\_end - data\_start >=10

);

--exista turisti care au achizitionat excursii cu o anumita durata, dar care au durat efectiv mai putin.

select ac.cod\_turist, ac.data\_end - ac.data\_start, e.durata, ac.cod\_excursie

from achizitioneaza ac join excursie e on (e.id\_excursie = ac.cod\_excursie)

where data\_end-data\_start <> durata ;

--cerinta proiect 12

--Implementarea a 3operații de actualizare sau suprimare a datelor utilizând subcereri.

create table achizitioneaza\_ib as select \* from achizitioneaza;

alter table achizitioneaza\_ib add (discount\_perioada number);

select ac.cod\_turist, ac.data\_end - ac.data\_start, e.durata, ac.cod\_excursie, ac.discount\_perioada

from achizitioneaza\_ib ac join excursie e on (e.id\_excursie = ac.cod\_excursie)

where data\_end-data\_start <> durata ;

--sa se modifice valoarea discount\_perioada astfel incat sa inregsitreze

--un procent de 1% pentru fiecare zi din durata excursiei care nu a fost achizitionata.

update achizitioneaza\_ib ac

set discount\_perioada = (select ac.data\_end-ac.data\_start - e.durata

from excursie e where (e.id\_excursie = ac.cod\_excursie)

)

where ac.data\_end-ac.data\_start < (select e.durata

from excursie e where (e.id\_excursie = ac.cod\_excursie)

) ;

--subcerere sincronizata

select \* achizitioneaza\_ib ac --diferenta de zile (durata efectiva durata planificata)

where discount\_perioada = (select ac.data\_end-ac.data\_start - e.durata

from excursie e where (e.id\_excursie = ac.cod\_excursie)

)

select ac.cod\_turist, ac.data\_end - ac.data\_start, e.durata, ac.cod\_excursie, ac.discount\_perioada

from achizitioneaza\_ib ac join excursie e on (e.id\_excursie = ac.cod\_excursie)

where data\_end-data\_start <> durata ;

create table emp\_ib as select \* from employees;

update emp\_ib e

set salary = (select e.salary + (e.salary-avg(salary))/2

from emp\_ib

where department\_id = e.department\_id )

where salary <= (select avg(salary)

from emp\_ib where department\_id = e.department\_id) ;

utilizarea a cel puțin 1 bloc de cerere(clauza WITH)

--sa se afiseze primele 2 excursii in fuctie de numarul de turisti care le-au achizitionat

select cod\_excursie, count(distinct data\_start)

from achizitioneaza

group by cod\_excursie;

with nr\_achizitioneaza as (

select cod\_excursie, count(distinct data\_start) nr

from achizitioneaza

group by cod\_excursie

)

select nr, cod\_excursie

from nr\_achizitioneaza

where nr in (

select nr from

(select nr, cod\_excursie

from nr\_achizitioneaza

order by nr desc)

where rownum <=2

);

--grupări de date, funcții grup, filtrare la nivel degrupuri

--Să se afişeze excursiile pentru care nu mai există locuri pentru data de plecare 14 -

--aug-2011.

select \* from excursie

where id\_excursie in (102, 201);

select cod\_excursie, count(distinct cod\_turist)

from achizitioneaza

where to\_char(data\_start, 'dd-mm-yyyy') = '14-08-2011'

group by cod\_excursie;

with nr\_locuri\_ocupate

as (select cod\_excursie, count(distinct cod\_turist) nr\_locuri

from achizitioneaza

where to\_char(data\_start, 'dd-mm-yyyy') = '14-08-2011'

group by cod\_excursie

)

select e.id\_excursie, e.pret

from excursie e join nr\_locuri\_ocupate n on (e.id\_excursie = n.cod\_excursie)

where e.nr\_locuri = n.nr\_locuri;

--Să se obŃină codurile turiştilor şi codul ultimei excursii achiziŃionate de către aceştia.

select t.id\_turist, e.id\_excursie

from turist t join achizitioneaza a on t.id\_turist=a.cod\_turist

join excursie e on e.id\_excursie=a.cod\_excursie

where a.data\_achizitie=(select max(data\_achizitie)

from turist join achizitioneaza

on id\_turist=cod\_turist

where id\_turist=t.id\_turist);

select ac.cod\_turist, ac.data\_achizitie

from achizitioneaza ac join

(select cod\_turist, max(data\_achizitie) mdata

from achizitioneaza

group by cod\_turist) max\_data

on (ac.cod\_turist = max\_data.cod\_turist and data\_achizitie = mdata);

with max\_data as (select cod\_turist, max(data\_achizitie) mdata

from achizitioneaza

group by cod\_turist)

select ac.cod\_turist, ac.data\_achizitie

from achizitioneaza ac join max\_data

on (ac.cod\_turist = max\_data.cod\_turist and data\_achizitie = mdata);

select d.department\_id, d.department\_name, avg(salary) medie

from employees e join departments d on (e.department\_id = d.department\_id)

group by d.department\_id, d.department\_name

--filtrare la nivel de grupuri

having avg(salary) = ( select max(medie) from (

select d.department\_id, d.department\_name, avg(salary) medie

from employees e join departments d on (e.department\_id = d.department\_id)

group by d.department\_id, d.department\_name)

);

with medie\_dep as (

select d.department\_id, d.department\_name, avg(salary) medie

from employees e join departments d on (e.department\_id = d.department\_id)

group by d.department\_id, d.department\_name

)

select \* from medie\_dep

where medie = ( select max(medie) from medie\_dep);

--L5. 26. Să se afişeze:

-- - suma salariilor, pentru job-urile care incep cu litera S;

-- - media generala a salariilor, pentru job-ul avand salariul maxim;

-- - salariul minim, pentru fiecare din celelalte job-uri.

select e.job\_id, j.job\_title,

CASE WHEN substr(e.job\_id, 1, 1) = 'S' THEN SUM(salary)

WHEN max(salary) = (select max(salary) from employees) THEN AVG(salary)

ELSE MIN(salary) END rezultat

from employees e join jobs j on (e.job\_id = j.job\_id)

group by e.job\_id, j.job\_title;

--

--Să se afişeze pentru fiecare agenŃie, denumirea şi profitul obŃinut.

--(Profitul obŃinut din vânzarea unei excursii este pret – pret \* discount

--Dacă discountul este necunoscut profitul este preŃul excursiei).

--sa se afiseze si agantiile cu profit 0

select a.id\_agentie, a.denumire, nvl(sum(e.pret - e.pret \* ac.discount), 0)

from agentie a left join excursie e on (e.cod\_agentie = a.id\_agentie)

left join achizitioneaza ac on (e.id\_excursie = ac.cod\_excursie)

group by a.id\_agentie, a.denumire;

pentru fiecare turist sa se afiseze nr e excursii achizitionate

de la agenŃii din ConstanŃa, de la agenŃii din Bucureşti.

select t.id\_turist, t.nume,

sum( decode (a.oras, 'Constanta', 1 , 0 ) ) nrc,

sum( decode (a.oras, 'Bucuresti', 1 , 0 ) ) nrb

from turist t join achizitioneaza ac on (t.id\_turist = ac.cod\_turist)

join excursie e on (e.id\_excursie = ac.cod\_excursie)

join agentie a on (a.id\_agentie = e.cod\_agentie)

group by t.id\_turist, t.nume;